AND SYSTEM FOR MULTIFRAME BUFFERING IN A ROUTING DEVICE," (Attorney Docket No. 030048025US); U.S. Patent Application No. _____ entitled "METHOD AND SYSTEM FOR DOMAIN ADDRESSING IN A COMMUNICATIONS NETWORK," (Attorney Docket No. 030048026US); U.S. Patent Application No. _____ entitled "METHOD AND SYSTEM FOR INTERSWITCH LOAD BALANCING IN A COMMUNICATIONS NETWORK," (Attorney Docket No. 030048027US); U.S. Patent Application No. _____ entitled "METHOD AND SYSTEM FOR INTERSWITCH DEADLOCK AVOIDANCE IN A COMMUNICATIONS NETWORK," (Attorney Docket No. 030048028US); U.S. Patent Application No. _____ entitled "METHOD AND SYSTEM FOR CONNECTION PREEMPTION IN A COMMUNICATIONS NETWORK," (Attorney Docket No. 030048029US); U.S. Patent Application No. _____ entitled "METHOD AND SYSTEM FOR MULTICASTING IN A ROUTING DEVICE," (Attorney Docket No. 030048030US); U.S. Patent Application No. _____ entitled "METHOD AND SYSTEM FOR NETWORK CONFIGURATION DISCOVERY IN A NETWORK MANAGER," (Attorney Docket No. 030048032US); U.S. Patent Application No. _____ entitled "METHOD AND SYSTEM FOR PATH BUILDING IN A COMMUNICATIONS NETWORK," (Attorney Docket No. 030048033US); U.S. Patent Application No. _____ entitled "METHOD AND SYSTEM FOR RESERVED ADDRESSING IN A COMMUNICATIONS NETWORK," (Attorney Docket No. 030048035US); U.S. Patent Application No. _____ entitled "METHOD AND SYSTEM FOR RECONFIGURING A PATH IN A COMMUNICATIONS NETWORK," (Attorney Docket No. 030048036US1); U.S. Patent Application No. _____ entitled "METHOD AND SYSTEM FOR ADMINISTRATIVE PORTS IN A ROUTING DEVICE," (Attorney Docket No. 030048037US); U.S. Patent Application No. _____ entitled "PARALLEL ANALYSIS OF INCOMING DATA TRANSMISSIONS," (Attorney Docket No. 030048038US); U.S. Patent Application No. _____ entitled "INTEGRATED ANALYSIS OF INCOMING DATA TRANSMISSIONS," (Attorney Docket No.

[0004] Existing enterprise data networks ("EDNs") that support e-commerce applications providing services to customers are straining under the demand to provide added performance and added services. The growing customer demands for services, along with a highly competitive market, has resulted in increasingly complex ad hoc EDNs. Affordable, high-performance EDN solutions require extensive scalability, very high availability, and ease of management. These attributes are significantly compromised or completely lost as existing solutions are grown to meet the demand.

[0005] Current architectures of EDNs typically include three sub-networks: 1) a local area network (LAN) for web and database servers, 2) a computational network for application servers, and 3) a storage area network (SAN). The processing and storage elements attached to these sub-networks may have access to a wide area network (WAN) or metropolitan area network (MAN) through a bridging device commonly known as an edge switch. Each of these sub-networks typically uses a distinct protocol and associated set of hardware and software including network interface adapters, network switches, network operating systems, and management applications. Communication through the EDN requires bridging between the sub-networks that requires active participation of server processing resources for protocol translation and interpretation.

[0006] There are many disadvantages to the current architecture of EDNs. The disadvantages result primarily because the multi-tiered architecture is fractured and complex. First, it is very difficult to integrate the disparate systems that use different communications protocols, interfaces, and so on. Second, overall performance suffers because each sub-network is managed separately, rather than being managed with comprehensive knowledge of the complete network. Third, the cost of maintaining three disparate types of network hardware and software can be high. Fourth, it is difficult to scale an architecture that uses such disparate systems. It would be desirable to have an architecture for EDNs that would be alleviate the many disadvantages of the current fractured multi-tiered architectures.

path from the source node to the destination node. The source node sends the data to its destination node by providing the data along with the virtual address to a routing device of the network. Upon receiving the data and the virtual address, a source-side port of each routing device in the path uses the virtual address to identify a destination-side port through which the data and the virtual address are to be transmitted. The network manager configures the routing devices by setting the mappings from a source-side port to a destination-side port for each routing device in the path. The routing devices receive data via source-side ports and transmits data via destination-side ports.

[0016] In one embodiment, the network manager may be centralized or distributed. A centralized network manager may reside at one node connected to the interconnect fabric. The centralized network manager may provide configuration information to the routing devices using in-band communications or out-of-band communications. In-band communications refers to the use of the communications links connecting the ports of the routing devices. Out-of-band communications refers to the use of communications links used specifically to connect the routing devices to the network manager. A centralized network manager may alternatively reside within a routing device. Each routing device may have the capabilities to be the network manager. Upon initialization, the routing devices may coordinate to select which of the routing devices is to function as the network manager. A distributed network manager, in contrast, may have its functions performed at various manager devices connected directly to the routing devices. The network manager at each manager device can control the routing device(s) to which it is directly connected. In addition, the network manager at each manager device can communicate with the network managers at other manager devices via in-band or out-of-band communications to coordinate control of the routing devices. In one embodiment, the distributed network manager can have different functions performed at various manager devices.

[0017] In one embodiment, the network manager identifies paths through the network from source nodes to destination nodes. The paths may be identified

initially when the network manager starts up, may be identified when the network topology (e.g., the routing devices of the network and their interconnections) changes (e.g., as a result of a failure), or may be identified dynamically when a registration request is received from a source node. One skilled in the art will appreciate that various combinations of these techniques for identifying a path may be used. For example, the network manager may identify paths dynamically at registration, but may re-identify paths when the topology of the network changes. Regardless of which of these techniques are used, the network manager would typically need to know the topology of the network to identify the paths.

[0018] In one embodiment, the network manager dynamically discovers the topology of the network at initialization. The network manager can discover the topology in several different ways. The network manager can be provided with configuration information that identifies the routing devices of the network. The network manager can use this configuration information to send a message to each routing device asking which of its ports are connected to another device. The network manager can then send a query message via each connected port asking the connected-to device to identify itself and its port. From the responses to the query messages, the network manager can identify the connections (i.e., communications links) between the routing devices and thus the topology of the network. Alternatively, rather than sending a query message to each connected-to port, the routing devices upon initialization can request the connected-to devices to provide their identifications. The routing devices can then provide the identifications of the connected-to ports to the network manager. The configuration information along with the identifications of the connected-to ports describes the network topology.

[0019] In another embodiment, the network manager can dynamically discover the identifications of the routing devices by sending query messages through the ports of the routing device to which it is directly connected. The network manager then becomes aware of each routing device that responds to the query. The

detect a failure along the existing path. The network manager may be able to use the same virtual address to configure the new path. If the network manager uses each virtual address only once, then the network manager can use the same virtual address for the new path. If, however, the same virtual address is used to identify different paths, then it may be possible that the configuration of the new path may conflict with the configuration of another path that uses the same virtual address. When the same virtual address can be used, then the network manager can change the path in a manner that is transparent to the source node. In particular, the network manager need not notify the source node of the change in the path. Also, if multiple destination nodes provide the same functionality, then the network manager may implement node load balancing by dynamically changing a path so that data will be sent to a different destination node. The use of these virtual addresses allows the changes to be made without changing the source and destination virtual addresses of the path.

[0025] In one embodiment, the network manager may reserve one or more virtual addresses for sending frames from a device (e.g., routing device or node) to the network manager. For example, such a frame may include a registration request from a source node. When the network manager is distributed, a routing device may detect when it has received a frame with a reserved virtual address and may forward the frame directly to the connected manager device for processing by the network manager. To provide flexibility, a frame directed to the network manager may include a combination of a reserved virtual address and another virtual address. When a routing device detects such a frame, it may determine whether it is configured to forward frames directed to the other virtual address using in-band communications. If so, the routing device forwards the frame through the destination-side port identified by the other virtual address. If the routing device is not configured for the other virtual address, then the routing device sends the frame to the network manager via out-of-band communications. For example, the routing device may send the frame to its directly connected manager device. In this way, the network manager can configure the network so that certain frames

that matches its domain address, then the frame has arrived at its destination domain The interconnect fabric module then forwards the frame in accordance with the destination virtual address since it has arrived at its destination domain. If, however, the domain addresses do not match, then the frame has not arrived at its destination domain. The interconnect fabric module forwards the frame using an inter-domain path. Each port of an interconnect fabric module may have a domain address table (configured by the network manager) that maps the domain addresses to the destination port through which frames with that domain address are to be forwarded. Thus, an interconnect fabric module may selectively use virtual addresses and domain addresses when forwarding frames.

[0028] In one embodiment, an interconnect fabric module uses a crosspoint switch to switch connect its source and destination ports. When the crosspoint switch has more switch ports than ports of the interconnect fabric module, the extra switch port can be used for administrative functions of the network manager. When an interconnect fabric module receives a frame directed to a virtual address reserved for administrative services of the network manager, the interconnect fabric module connects the source port to the extra switch port which is connected to a manager device. When the frame is transmitted from the source port, the network manager at the manager device receives the frame and processes it in accordance with its administrative functions. In this way, administrative frames can be directly forwarded to the network manager when they are first received by an interconnect fabric module from a node.

[0029] In some embodiments, one or more virtual identifier ("VI") Network Interface Controller ("NIC") facilities on each node (e.g., one VI NIC for each network interface) facilitate the use of virtual identifiers in communicating data When a VI NIC on a node receives an indication that a data communication to one or more remote nodes is to occur, such as from an application executing on the node, the VI NIC will identify an appropriate transmittal virtual identifier that can be used to route the data communication through the network to the appropriate remote destination nodes without being assigned to or directly

associated with those destination nodes. Such data communications can include both transitory connectionless transmittals of data (e.g., unidirectional transmittals from a source to a destination) and non-transitory connections that allow multiple distinct transmittals of data (e.g., a persistent dedicated connection that allows a connection-initiating source and a connection destination to transmit data back and forth).

[0030] The VI NIC can identify an appropriate transmittal virtual identifier for routing a data communication in various ways. In some embodiments, the VI NIC will register some or all outgoing data communications with a network manager for the network, and will receive an appropriate transmittal virtual identifier to be used for that communication from the network manager. If an indicated data communication corresponds to a previously registered data communication (e.g., to an existing connection or to a previous communication to the same destination and in the same transmission manner), however, the VI NIC could instead in some embodiments use the previously received transmittal virtual identifier for that data communication rather than perform an additional registration for the indicated data communication. The manners in which a data communication can be transmitted vary with the transmission characteristics that are supported by a network, and can include factors such as a particular Class Of Service ("COS") or transmission priority.

[0031] In some embodiments, when a data communication indicated by a source can result in bi-directional communication (e.g., a response from one or more of the destinations), the VI NIC also identifies a response virtual identifier that can be used for routing data from one or more of the destinations back to the source. If the VI NIC registers the data communication with a network manager, this response virtual identifier may be received from the network manager. After identifying this response virtual identifier, the VI NIC associates it with information indicating how to process received data communications that are routed using the response virtual identifier. In some embodiments, such received data communications are processed by forwarding the data communications to one or

connected to other devices. The network manager uses this information to send a message through each port that is connected to another device to identify the connected-to device. Figure 2 is a flow diagram illustrating the discovery processing of a component of the interconnect fabric module in one embodiment. Each port of an interconnect fabric module identifies whether it is connected to a port of another device, such as another switch or a node. The interconnect fabric module then provides to the network manager an indication of which of its ports are connected to other ports to assist in the discovery process. In blocks 201-204, the component determines whether each port is currently connected to another port. In block 201, the component selects the next port. In decision block 202, if all the ports have already been selected, then the component completes, else the component continues at block 203. In decision block 203, the component determines whether the selected port is connected to another port. This determination may be made based on various voltage levels of the communications links. If there is a connection, then the component continues at block 204, else the component loops to block 201 to select the next port of the interconnect fabric module. In block 204, the component notes the selected port as connected to another port and loops to block 201 to select the next port of the interconnect fabric module.

[0035] Figure 3 is a flow diagram illustrating the discovery processing of the network manager in one embodiment. The network manager first retrieves an indication of which ports of the interconnect fabric modules are connected to other devices. The network manager then sends a query message through each of the indicated ports to the connected-to port. When the connected-to port receives the query message, it responds with an identification of its interconnect fabric module and its port number. In this way, the network manager can discover the topology of the interconnect fabric. In blocks 301-303, the network manager retrieves the indications of which ports of the interconnect fabric modules are connected to other ports. In block 301, the network manager selects the next interconnect fabric module that has not yet been selected. In decision block 302, if all the

direction. One skilled in the art will appreciate that various well-known techniques for identifying paths can be used. In block 402, the component invokes an identify virtual address component passing an indication of the path and an indication that the virtual address to be used by the source node when sending a communications to the destination node (e.g., the destination virtual address). The invoked component may select a virtual address that is not currently in use by any of the source-side ports of the path. A source-side port of the path is a port that receives data sent by a source node, and a destination-side port of the path is a port through which data is transmitted on its way to the destination node. In block 403, the component invokes in identify virtual address component passing an indication of the path and that the virtual address is to be used by the destination node (e.g., the source virtual address). In block 404, the component invokes a component to initialize the label tables of the source-side ports of the path with the destination virtual address. The invoked component transmits instructions to the each source-side port of the path indicating that the port is to update its label table to map the source virtual address to a destination-side port of the interconnect fabric module. In block 405, the component invokes a component to initialize the label tables of the destination-side ports of the path with the source virtual address. The component then completes.

[0038] Figure 5 is a flow diagram illustrating the processing of an identify virtual address component of the network manager in one embodiment. In this embodiment, the identify virtual address component is provided an indication and a path along with an indication of whether a virtual address for the source node or the destination node is to be identified. The component may check every port along the path to identify a virtual address that is not currently used by a port along the path. Alternatively, the component may identify virtual addresses based on a sequential ordering. That is, the component may keep track of the last identified virtual address and increment that virtual address to identify the next virtual address. In this way, each virtual address is unique. In blocks 501-505, the component loops selecting the next virtual address and determining whether it

Reserved Addressing

[0040] In one embodiment, the crosspoint switch of an IFM may have more outputs than the number of ports of the IFM. For example, a crosspoint switch may have 34 inputs and outputs, but the IFM may have only 32 ports. The IFM may use these additional ports of the crosspoint switch to route upper layer protocol frames, such as frames directed into a name server or other administrative services. In one embodiment, the additional output ports of the crosspoint switch may be connected to a manager device for the IFM. An interconnect fabric module may have a list of "reserved" addresses that designate an upper layer protocol port. When an IFM determines that an address of its frame matches one of the reserved addresses, it enables the routing of that frame to an upper layer protocol port. The routing to upper layer protocol ports may use the same arbitration mechanism as used for routing to non-upper layer protocol ports as described in the Patent Application entitled "Interconnect Fabric Module." Alternatively, when the crosspoint switch does not have extra output for an upper layer protocol port, an output can be selectively switched between a communications port and an upper layer protocol port depending on whether the address of the destination identifier is reserved.

[0041] Figure 7 is a block diagram illustrating a distributed network manager in one embodiment. In this embodiment, the network manager may be implemented on a series of manager devices connected directly to the interconnect fabric modules. The distributed network manager may communicate with each other using in-band communication of the interconnect fabric or using out-of-band communication that is independent of the interconnect fabric The crosspoint switch of an interconnect fabric module may have reserved ports for the distributed network manager. When an interconnect fabric module receives data that designates one of the reserved ports, then the interconnect fabric module forwards the data to the distributed network manager through the reserved port.

[0042] Figure 8 is a flow diagram illustrating the processing of a component of an interconnect fabric module that processes reserved addresses in one